

---

# Red Hat Network

## Satellite 5.3.0

### ISS Best Practices Whitepaper

Red Hat Network Satellite

Pradeep Kilambi  
Miroslav Suchy

Copyright © 2009 Red Hat, Inc. This material may only be distributed subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version of the OPL is presently available at <http://www.opencontent.org/openpub/>).

Red Hat and the Red Hat "Shadow Man" logo are registered trademarks of Red Hat, Inc. in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

1801 Varsity Drive  
Raleigh, NC 27606-2072 USA  
Phone: +1 919 754 3700  
Phone: 888 733 4281  
Fax: +1 919 754 3701  
PO Box 13588 Research Triangle Park, NC 27709 USA

#### Abstract

Red Hat Network Satellite 5.3.0 supports direct synchronization of channels from one Satellite to other. This guide will help you understand and make the most of Inter-Satellite Sync (ISS), guiding you through some basic best practices in a task-oriented manner.

1. Required Environment for ISS .....	2
2. Recommended Models for Inter-Satellite Sync .....	2
3. Configuring the Master RHN Satellite Server .....	3
4. Configuring the Slave RHN Satellite Servers .....	4
5. Using Inter-Satellite Sync .....	4
5.1. Syncing between a Development Staging Server and a Production Satellite .....	5
5.2. Bi-directional sync .....	5
6. Synchronizing by Organization .....	6
<b>A. Revision History</b>	<b>6</b>

RHN Satellite 5.3.0 supports synchronization between two Satellites. This synchronization, also called *Inter-Satellite Sync*, allows administrators to simplify the process of coordinating content from one RHN Satellite source to another or several others.

## 1. Required Environment for ISS

The following are the basic requirements for Inter-satellite Sync.

- At least two RHN Satellite 5.3.0 servers
- At least one RHN Satellite populated with at least one channel
- Master RHN Satellite SSL certificate available on each of the slave RHN Satellites for secure connection

## 2. Recommended Models for Inter-Satellite Sync

The Inter-Satellite Sync feature for Satellite provides facilities for synchronizing content between two or more Satellites. The following are some of the more typical uses that show the possibilities of Inter-Satellite Sync and help guide you in determining how to make the most of this feature in your environment.

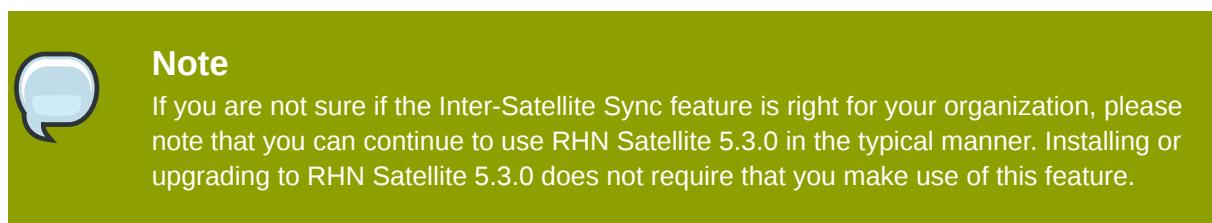


Figure 1. Staging Satellite

In this example, the Stage Satellite is used to prepare the content and perform quality assurance (QA) work — to make sure that packages are fit for production use. After content is approved to go to production, the Production Satellite will then synchronize the content from the Stage Satellite.

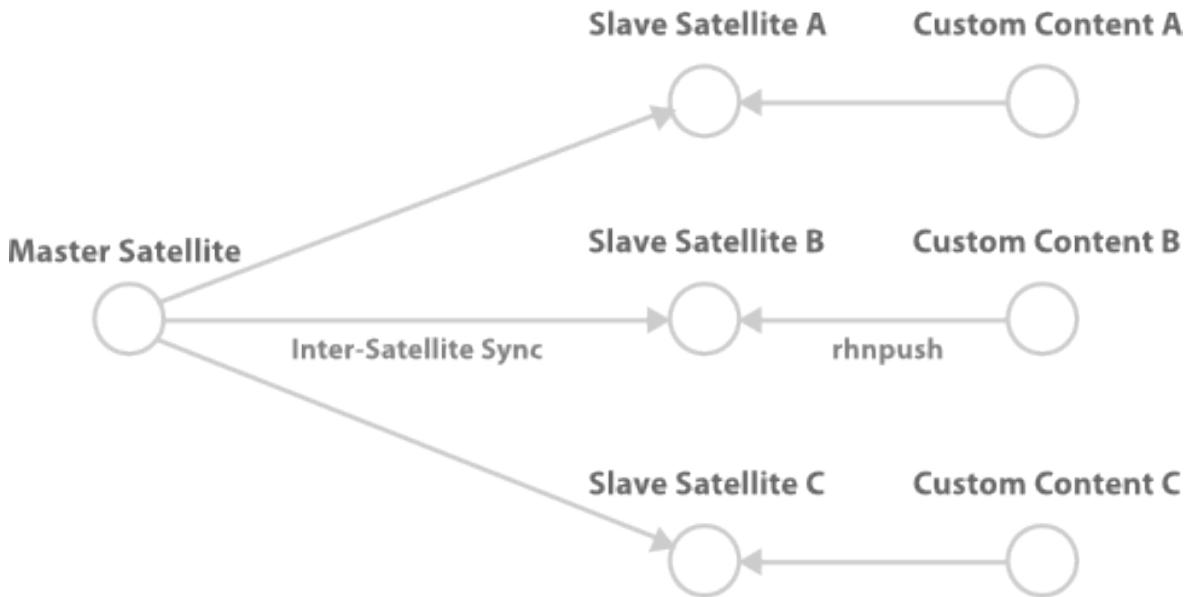


Figure 2. Master Server and Slave Peers that include their own custom content

In this example, the master Satellite is the development channel, from which content is distributed to all production slave Satellites. Some slave Satellites have extra content not present in master Satellite channels. These packages are preserved, but all changes from master Satellite are synchronized to Slave Satellite.

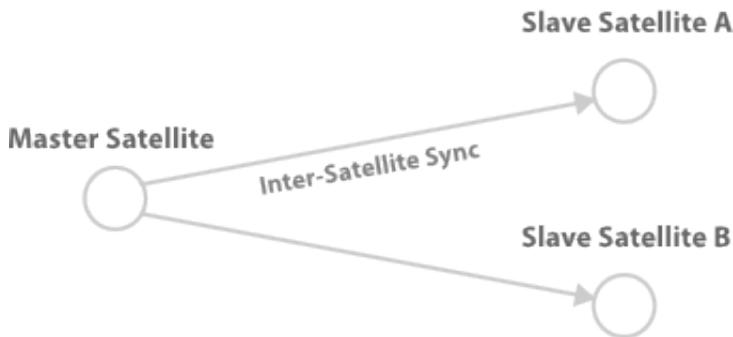


Figure 3. Slave Satellites are maintained exactly as the master Satellite

In this example, the master Satellite (for example, a software or Hardware vendor) provides data directly to its customer. These changes are regularly synchronized to slave Satellites.

### 3. Configuring the Master RHN Satellite Server

To use the inter-satellite sync feature, you must first ensure that you have it enabled. Make sure that the `/etc/rhn/rhn.conf` contains the following line:

```
disable_iss=0
```

In the same file is the variable:

```
iss_rhn=1
```

```
allowed_iss_slaves=
```

By default, no slave Satellites are specified to sync from the master server, so you must enter the hostname of each slave Satellite server, separated by commas. For example:

```
allowed_iss_slaves=slave1.satellite.example.org,slave2.satellite.example.org
```

Once you finished configuring the **rhn.conf** file, restart the **httpd** service by issuing the following command:

```
service httpd restart
```

## 4. Configuring the Slave RHN Satellite Servers

To configure RHN Satellite slave servers for Inter-Satellite Sync, ensure that you have the ORG-SSL certificate from your master RHN Satellite server so you can securely transfer content. This can be downloaded over http from the **/pub/** directory of any Satellite. The file is called **RHN-ORG-TRUSTED-SSL-CERT**, but can be renamed and placed anywhere on the slave Satellite, such as the **/usr/share/rhn/** directory.

For information about SSL configuration for use with RHN Satellite, refer to Chapter 3, "SSL Infrastructure" in the *RHN Satellite Client Configuration Guide*.

Once the SSL certificate is placed on the slave server, you can see the list of channels available to sync from the master Satellite server by running the following command (replacing the **master.satellite.example.com** with the hostname of the master Satellite server):

```
satellite-sync --iss-parent=master.satellite.example.com --ca-cert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT --list-channels
```

This command lists both Red Hat Network Hosted channels as well as any custom channels available on the master Satellite server.

## 5. Using Inter-Satellite Sync

Now that Inter-Satellite Sync is configured, you can now use it to synchronize channels from the master Satellite to the slave servers.

On the slave servers, configure the Master server hostname and SSL certificate file path in the following lines of the **/etc/rhn/rhn.conf** file:

```
iss_parent      = master.satellite.domain.com
iss_ca_chain   = /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT
```

Then run the **satellite-sync** command by typing:

```
satellite-sync -c your-channel
```

### Note

Any command line options to the **satellite-sync** command will override any default or customized settings in the **/etc/rhn/rhn.conf** file.

## 5.1. Syncing between a Development Staging Server and a Production Satellite

There may be instances where an administrator wants to sync data from a staging server that has custom channels that are ready for production use to a production Satellite server.

For example, a production Satellite Server normally syncs directly from RHN Hosted servers for content updates, but will occasionally sync production-ready information from a RHN Satellite development server.



Figure 4. Syncing from RHN Hosted and a Satellite Staging Server

Normally, the administrator runs:

```
satellite-sync -c your-channel
```

This command downloads directly from data from **rhn\_parent** (usually RHN Hosted, rhn.redhat.com). Then, to sync from the staging Satellite server address , the administrator runs:

```
satellite-sync --iss-parent=staging-satellite.example.com -c custom-channel
```

## 5.2. Bi-directional sync

Administrators can configure an environment where two RHN Satellite servers act as masters of each other. For example, Satellite A and B can sync content from one another.



Figure 5. Bi-directional syncing

Both Satellites would need to share SSL certificates, then set the **iss\_parent** option in the **/etc/rhn/rhn.conf** file of Satellite A to point to the hostname of Satellite B, and do the same for Satellite B to point to Satellite A as the **iss\_parent**.

## 6. Synchronizing by Organization

Satellite-sync has a new enhancement as part of the inter satellite sync feature where a user can import content to any specific organization. This can be done locally or by a remote syncing from hosted or another satellite.

The aim is for Satellite sync to be able to import content with respect to org\_id. This targets two sets of users. One is the disconnected Multi-Org case, where the main source of content for the user is either to get content from channel dumps or to export them from connected satellites and import it to the Satellite. The user mainly hosts custom channels from disconnected satellites. If they wish to export custom channels from connected satellites, they can do so by organizational sync.

The other case is a connected Multi-Org satellite customer. These new flags could work as a means of moving content between multiple orgs.

Synchronizing by organization has a few rules that it follows to maintain the integrity of the source org.

- If the source content belongs to a base org (any Red Hat content) it will default to the base org even if a destination org is specified. This ensures that the specified content is always in that privileged base org.
- If an org is specified at the command line, it will import content from that org.
- If no org is specified, it will default to org 1.

The following are three example scenarios where organizational IDs (orgid) are used to synchronize between Satellites:

1. Import content from master to slave satellite.

```
satellite-sync --parent-sat=master.satellite.domain.com -c channel-name  
--orgid=2
```

2. Import content from an exported dump of a specific org

```
$ satellite-sync -m /dump -c channel-name --orgid=2
```

3. Import content from RHN Hosted (assuming the system is registered and activated. If the source org is not specified, the base Red Hat channel is chosen).

```
$ satellite-sync -c channel-name
```

## A. Revision History

Revision 1.0